Parameter Efficient Fine-tuning via Cross Block Orchestration for Segment Anything Model

Zelin Peng[1,†], Zhengqin Xu[1,†], Zhilin Zeng[1], Lingxi Xie[2], Qi Tian[2], and Wei Shen[1(✉)]

[1]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[2]Huawei Inc.

{zelin.peng, fate311, bernardeschi, wei.shen}@sjtu.edu.cn;
198808xc@gmail.com; tian.qi1@huawei.com

Abstract

Parameter-efficient fine-tuning (PEFT) is an effective methodology to unleash the potential of large foundation models in novel scenarios with limited training data. In the computer vision community, PEFT has shown effectiveness in image classification, but little research has studied its ability for image segmentation. Fine-tuning segmentation models usually requires a heavier adjustment of parameters to align the proper projection directions in the parameter space for new scenarios. This raises a challenge to existing PEFT algorithms, as they often inject a limited number of individual parameters into each block, which prevents substantial adjustment of the projection direction of the parameter space due to the limitation of Hidden Markov Chain along blocks. In this paper, we equip PEFT with a cross-block orchestration mechanism to enable the adaptation of the Segment Anything Model (SAM) to various downstream scenarios. We introduce a novel inter-block communication module, which integrates a learnable relation matrix to facilitate communication among different coefficient sets of each PEFT block's parameter space. Moreover, we propose an intra-block enhancement module, which introduces a linear projection head whose weights are generated from a hyper-complex layer, further enhancing the impact of the adjustment of projection directions on the entire parameter space. Extensive experiments on diverse benchmarks demonstrate that our proposed approach consistently improves the segmentation performance significantly on novel scenarios with only around 1K additional parameters.

1. Introduction

A notable recent development in AI community is large foundation models [4, 20], which have made an increas-
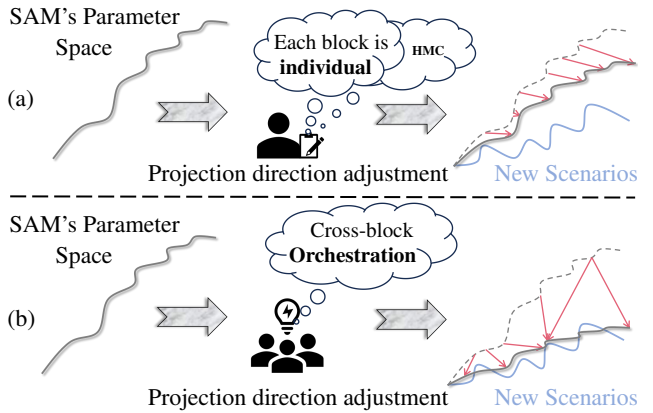

Figure 1. **Comparison between traditional PEFT paradigms and our proposed SAM-COBOT.** (a) Traditional methods typically adjust the projection direction of each layer in SAM's parameter space individually, which is limited by the Hidden Markov Chain (HMC). This often leads to relatively minor adjustments. (b) In contrast, our SAM-COBOT approach enhances PEFT with cross-block orchestration, enabling more effective and large adjustments of the projection directions.

ing impact widely across various domains, e.g., natural language processing (NLP) [17] and computer vision [10]. Despite their surprising zero-shot performance, fine-tuning still remains a crucial step for unleashing their potential in novel scenarios [26, 28]. To mitigate the extensive fine-tuning costs associated with a large number of pre-trained parameters, parameter efficient fine-tuning (PEFT) [6, 13, 14], which involves tuning a small subset of parameters while maintaining the vast majority frozen, has attracted increasing attentions in various fields.

In the field of computer vision, the majority of leading PEFT methodologies have focused on image classification tasks [15, 16, 40]. These approaches demonstrate that large classification models can be fine-tuned by injecting a small number of parameters. As evidence, their perfor-

---

✉ Corresponding Author: wei.shen@sjtu.edu.cn
† Indicates equal contribution.

mance can match or even surpass that of vanilla full fine-tuning [15]. Influenced by the success of the existing PEFT methods, one can directly apply them in fine-tuning segmentation models [26, 39], e.g., Segment Anything Model (SAM) [20]. However, fine-tuning segmentation models often necessitates a heavier adjustment of parameters as the output space for segmentation is often much larger and more varied compared to classification, which poses a challenge for current PEFT methods that are limited by the Hidden Markov Chain (HMC) along layers [21, 41]. The memoryless nature of HMC implies that each layer's state is influenced only by its adjacent layers, thus readily leading to minor adjustments of the projection directions in the entire parameter space for new scenarios, as shown in Fig. 1.

In this paper, we equip PEFT with a **Cr**O**ss-B**Lock **O**rches**T**ration mechanism to enable the adaptation of SAM to various downstream scenarios, dubbed as SAM-COBOT. The goal of SAM-COBOT is to explicitly integrate cross-block orchestration to enhance the flexibility and reliability of adjusting projection directions. Specifically, we first propose an inter-block communication (IBC) module, which introduces a learnable relation matrix to capture interdependence and facilitate communication among different blocks. The communication is realized by adjusting the coefficient set of each PEFT block's parameter space. We treat all the coefficient sets of PEFT block's parameter space as a tensor, and use the learnable relation matrix to capture cross-slice information for adjusting each coefficient set. Consequently, IBC allows projection directions to influence each other in the entire parameter space. Subsequently, we introduce an intra-block enhancement (IBE) module, which includes a linear project head whose weights are generated from a hyper-complex layer, to ensure that any coordinated adjustments made to the projection directions achieve a greater impact on the entire parameter space.

Extensive experiments show that the proposed SAM-COBOT can be easily plugged-and-play and consistently improve various PEFT paradigms, e.g., LoRA [14] and Adaptformer [6] by a large margin across three prevalent scenarios in computer vision, including natural image segmentation, remote sensing image segmentation, and medical image segmentation. Additionally, SAM-COBOT only needs to introduce around **1K** parameters (using ViT-Base [10] as the backbone) while achieving superior segmentation performance.

2. Related Work

Parameter Efficient Fine-tuning. The objective of parameter-efficient fine-tuning (PEFT) is to utilize the weights from a pre-trained network and tune them for the downstream task by introducing a minimal number of trainable parameters. Many PEFT methods [31, 32, 40, 44] (in deep learning era) are believed to be derived from

Adapter [33] which introduces a few modules into the pre-trained network. After that, numerous efforts have been devoted to improving the pipeline. For example, some tailored modules like Adapterformer [6] and RepAdapter [24] are designed for different vision tasks, e.g., classification. Recently, since LoRA [14], which replaces additional modules by introducing low-rank matrices to alter the initial parameter spaces, attained growing attention, some other works focusing on automating matrix engineering [16, 47] to boost the LoRA structure. Apart from the above, some studies also attempt to break new ground, e.g., add extra parameters as prompts along with the inputs [15], scales and shifts features after each transformer block [22], fine-tune the bias terms in each layer [45], to name a few.

Considering the increasing attention of the large foundation model, i.e., SAM, this paper studies how to boost existing PEFT techniques for fine-tuning SAM [20], from a fresh viewpoint: cross-block orchestration.

Cross-block Orchestration. As there exist complementary learning patterns among different blocks (i.e., the shallow block features preserve more details while the deeper one captures more semantics), cross-block orchestration becomes a crucial component of recent state-of-the-art visual recognition algorithms [25, 42, 43, 49]. For example, CLR-Net [50] presented a cross-block refinement module to fully utilize both high-level and low-level features. Zhang *et al.* [46] introduced several self-regulation losses to fully understand detailed features and visual contexts. Chen *et al.* [5] proposed an adaptive cross-block correlation to recognize the style of visual arts.

This paper also refers to cross-block orchestration. The key differences are that (1) We address the restraint of the Hidden Markov Chain by orchestrating in the parameter space, as opposed to the traditional feature space and (2) for the first time, we introduce a hyper-complex layer [29] to facilitate approaching proper projection directions.

3. Preliminaries

3.1. Hyper-complex Number

In mathematics, hyper-complex number is a traditional term for an element of a finite-dimensional unital algebra over the field of real numbers. Its elements are generated with real number coefficients $(a_0, \cdots, a_n)$ for a basis $\{1, j_1, \cdots, j_n\}$. A n-dimensional hyper-complex number is defined in a n-dimensional space as:

$$h = a_0 1 + a_1 j_1 + a_2 j_2 + \cdots + a_{n-1} j_{n-1}. \quad (1)$$

In the hyper-complex number, a_0 is the real part, $a_1 j_1 + a_2 j_2 + \cdots + a_{n-1} j_{n-1}$ is the imaginary part. For simplicity, we consider $n = 4$ in this work. The imaginary part of a 4-dimensional hyper-complex number satisfies:

$$j_1^2 = j_2^2 = j_3^2 = j_1 j_2 j_3 = -1. \quad (2)$$

The geometric interpretation of j_1, j_2, and j_3 can be understood as rotations in $\mathbb{R}^3$ space: the j_1 rotation represents the rotation from the X-axis to the Y-axis in the plane intersecting both axes, the j_2 rotation represents the rotation from the Z-axis to the X-axis in the plane intersecting both axes, and finally, the j_3 rotation represents a rotation from Y-axis to Z-axis in a plane that intersects both axes. The negative counterparts $-j_1$, $-j_2$, and $-j_3$ represent reverse rotations of their respective positive counterparts. It is worth noting that, unlike the real and complex numbers, multiplication of the imaginary part of the 4-dimensional hyper-complex number is not commutative, for example, $j_1 j_2 = j_3$ and $j_2 j_1 = -j_3$.

3.2. Segment Anything Model (SAM)

Segment Anything Model (SAM) [20] mainly consists of an image encoder characterized by a vast parameter set, followed by a lightweight mask decoder. The image encoder is structured with L sequential transformer layers. Besides, SAM also incorporates a dedicated prompt encoder, which adaptively handles both dense (mask-based) and sparse (box or point-based) prompts.

3.3. Parameter Efficient Fine-tuning (PEFT)

Problem Formulation. Given a large foundation model $\mathcal{F}$, e.g., SAM [20], the goal of PEFT is to fine-tune $\mathcal{F}(\mathbf{X}; \boldsymbol{\omega})$ to enable the foundation model to adapt to a new downstream task, where $\mathbf{X}$ is an input image from a dataset of the new task and $\boldsymbol{\omega} \in \boldsymbol{\Omega} \backslash \boldsymbol{\Omega}_l$ denotes the parameters of PEFT modules, which are trainable, while $\boldsymbol{\Omega}_l$ refers to the parameter set of $\mathcal{F}$, which are often frozen. Accordingly, the objective function of PEFT is formulated as:

$$\boldsymbol{\omega}^* = \arg \min_{\boldsymbol{\omega}} \mathcal{L}(\mathbf{X}, \mathbf{Y}), \tag{3}$$

where $\mathbf{Y}$ is a full dense label map. For segmentation tasks, the loss function $\mathcal{L}$ is commonly selected as a cross-entropy loss. We here briefly review two representative PEFT methods in SAM's adaptation, i.e., Adaptformer [6] and LoRA [14], since our method is based on them.

Adapterformer. Adapterformer [6] introduces a parallel learnable branch for the MLP module in each transformer layer. This branch is primarily composed of a down-projection layer characterized by parameters $\boldsymbol{\omega}_{down} \in \mathbb{R}^{D \times V}$ and an up-projection layer represented by parameters $\boldsymbol{\omega}_{up} \in \mathbb{R}^{V \times K}$. Here, the hidden dimension H is significantly smaller than the minimum of D and K, i.e., $H \ll \min(D, K)$. The value of H determines the size of the newly introduced parameter space.

LoRA. LoRA [14] introduced a learnable low-rank matrix $\boldsymbol{\omega}$ that works in parallel to the original weight matrix $\boldsymbol{\omega} \in \mathbb{R}^{D \times K}$, which is frequently associated with the parameter matrix in the multi-head self-attention module of

each transformer layer. $\boldsymbol{\omega}$ is derived by a QR decomposition, denoted as $\boldsymbol{\omega} = \boldsymbol{\beta}\boldsymbol{\alpha}$, where $\boldsymbol{\beta} \in \mathbb{R}^{D \times V}$, $\boldsymbol{\alpha} \in \mathbb{R}^{V \times K}$.

Drawbacks. Although existing PEFT methods can be directly integrated with SAM, we notice an opening question in this intuitive solution. Fine-tuning segmentation models often necessitates a heavier adjustment of parameters to align projection directions in the parameter space for new scenarios compared to classification models. However, these PEFT methods introduce only a limited number of parameters in each layer, which can only make relatively small adjustments of projection directions due to the limitation of Hidden Markov Chain (HMC) along SAM's layers. Although some methods, e.g., LST [38], seem to bypass this issue by integrating a learnable side adapter, the updating of each layer in the side adapter is also limited to interactions with its adjacent layers. Consequently, the limitations of HMC still exist. In contrast, we devise a plug-and-play method which directly mitigates the constraint of the HMC for existing PEFT methods while introducing nearly zero training efforts.

4. Methodology

4.1. Overview of SAM-COBOT

Fig. 2 illustrates the proposed SAM-COBOT framework, which equips PEFT with cross-block orchestration. Each PEFT block's parameter space comes from three parts, (1) PEFT module, (2) Inter-block communication module, and (3) Intra-block enhancement module.

4.2. Inter-block Communication

Coefficient Set Generation. Following previous studies [30, 47], the parameter space of a PEFT block $\boldsymbol{\omega}$ can be decomposed into a base set and a coefficient set. Considering the significantly larger number of base parameters compared to those of coefficients, we opt to facilitate inter-block communication through the coefficient set associated with each block. To this end, we first introduce a learnable diagonal matrix $\boldsymbol{\Lambda}$, which is defined as follows:

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_V \end{bmatrix} \in \mathbb{R}^{V \times V}, \tag{4}$$

where the diagonal elements $\{\lambda_i\}$ comprise a set of coefficients.

After that, we propose to introduce a learnable relation matrix for achieving inter-block communication. To do this, we need to conceptualize all coefficient set matrices in the entire parameter space as a single tensor. Denote the number of transformer blocks in SAM's image encoder as L, and each block contains a PEFT module with a coefficient set that we proposed. We treat the diagonal matrix of each
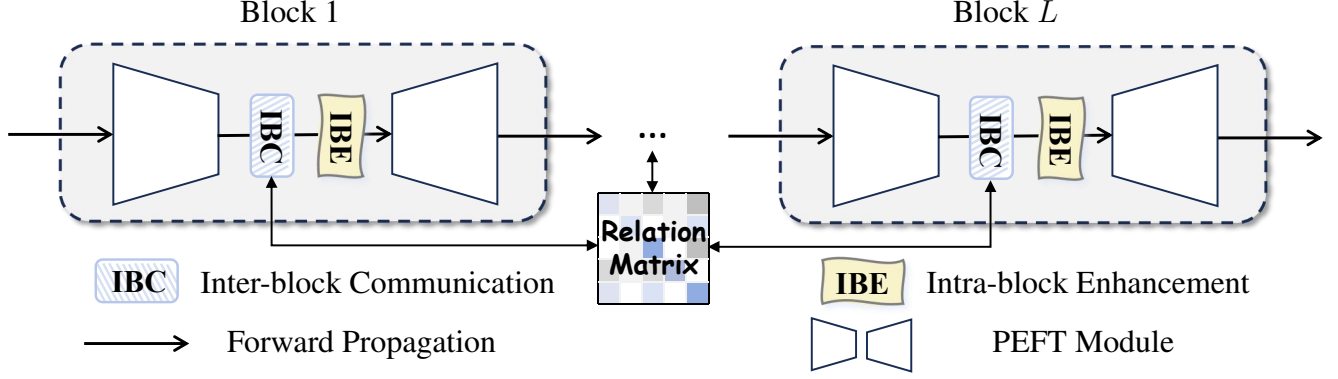
Figure 2. **A schematic representation of SAM-COBOT.** In the SAM-COBOT framework, we integrate an inter-block communication module followed by an intra-block enhancement module in each PEFT block.
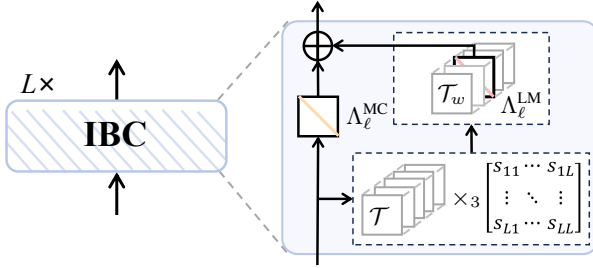

Figure 3. **The detailed structure of inter-block communication (IBC) module.** We introduce two coefficient sets, Λ_ℓ^{MC} and Λ_ℓ^{LM}, the former is communicated under the limitation of HMC, and the latter communicates with other coefficient sets among different blocks. (Best viewed in color).

coefficient set as an individual slice in this tensor $\mathcal{T}$, which is derived as:

$$\mathcal{T} = [\boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2, \cdots, \boldsymbol{\Lambda}_L] \in \mathbb{R}^{V \times V \times L}. \quad (5)$$

Then, according to the characteristics of gradient propagation in deep learning theory, i.e., chain rule, each frontal slice $\boldsymbol{\Lambda}_i \in \mathbb{R}^{V \times V}$ of the tensor $\mathcal{T} \in \mathbb{R}^{V \times V \times L}$ is updated sequentially, and thus update the tensor $\mathcal{T}$ is often slow. To avoid the cross-frontal-slice information loss in the tensor $\mathcal{T}$ during learning, we introduce the idea of a special tensor product, i.e., **T-product**.

Definition 4.1. (T-product) For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times l \times n_3}$, the T-product $\mathcal{C} \in \mathbb{R}^{n_1 \times l \times n_3} = \mathcal{A} * \mathcal{B}$ is defined as:

$$\mathcal{C} = \mathcal{A} * \mathcal{B} = \texttt{fold}(\texttt{bcirc}(\mathcal{A}) \cdot \texttt{unfold}(\mathcal{B})), \quad (6)$$

where

$$\texttt{bcric}(\mathcal{A}) = \begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{A}^{(n_3)} & \cdots & \mathbf{A}^{(2)} \\ \mathbf{A}^{(2)} & \mathbf{A}^{(1)} & \cdots & \mathbf{A}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{(n_3)} & \mathbf{A}^{(n_3-1)} & \cdots & \mathbf{A}^{(1)} \end{bmatrix}, \quad (7)$$

$$\texttt{unfold}(\mathcal{A}) = [\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \cdots, \mathbf{A}^{(n_3)}]^T, \quad (8)$$

$$\texttt{fold}(\texttt{unfold}(\mathcal{A})) = \mathcal{A}, \quad (9)$$

$\mathbf{A}^{(i)}$ denotes the i-th frontal slice $\mathcal{A}(:,:,i)$ of $\mathcal{A}$. There is an invertible linear transform $S: \mathbb{R}^{n_1 \times n_2 \times n_3} \to \mathbb{R}^{n_1 \times n_2 \times n_3}$ and it transforms the Eq. (6) as

$$\mathcal{C} = \texttt{S}^{-1}(S(\mathcal{A}) \odot \texttt{S}(\mathcal{B})) = \texttt{S}^{-1}(\bar{\mathcal{A}} \odot \bar{\mathcal{B}}) = \texttt{S}^{-1}(\bar{\mathcal{C}}), \quad (10)$$

where $\bar{\mathcal{C}} = \bar{\mathcal{A}} \odot \bar{\mathcal{B}}$ denotes the frontal-slice-wise product (Definition 2.1 refers to [18]) $\bar{\mathbf{C}}^{(i)} = \bar{\mathbf{A}}^{(i)}\bar{\mathbf{B}}^{(i)}, i = 1, 2, \cdots, n_3$. According to the definition of the frontal-slice-wise product, the invertible linear transform S is formulated as:

$$\bar{\mathcal{A}} = \texttt{S}(\mathcal{A}) = \mathcal{A} \times_3 \mathbf{S}, \quad (11)$$

where "$\times_3$" denotes the mode-3 product and $\mathbf{S} \in \mathbb{R}^{n_3 \times n_3}$ is an arbitrary invertible matrix. Similarly, the inverse transform of Eq. (11) is derived as:

$$\mathcal{A} = \texttt{S}^{-1}(\bar{\mathcal{A}}) = \bar{\mathcal{A}} \times_3 \mathbf{S}^{-1}. \quad (12)$$

Derivation. please refer to supplementary material.

■

According to Eqs. (10), (11), and (12), we adopt its idea and design an arbitrary invertible relation matrix $\mathbf{S} \in \mathbb{R}^{L \times L}$ to capture the cross-slice information in $\mathcal{T}$. Then the whole tensor $\mathcal{T}_w$ is formulated as:

$$\begin{aligned} \mathcal{T}_w &= \mathcal{T} \times_3 \mathbf{S} \\ &= [\boldsymbol{\Lambda}_1^l, \boldsymbol{\Lambda}_2^l, \cdots, \boldsymbol{\Lambda}_L^l] \in \mathbb{R}^{V \times V \times L}, \end{aligned} \quad (13)$$
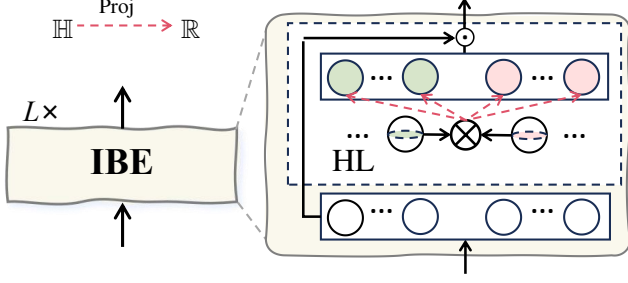
Figure 4. **The detailed structure of intra-block enhancement (IBE) module.** We introduce a hyper-complex layer (HL) for facilitating communication among projection directions in each layer. "Proj": Projection. "HL": hyper-complex layer. $\mathbb{H}$: hyper-complex space, i.e., suprasphere. (Best viewed in color) "$\otimes$": `Hamilton product`.

where $\times_3$ denotes mode-3 product and the relation matrix $\mathbf{S}$ is learnable.

Dual Coefficient Sets. Although HMC limits the substantial adjustment of parameter space, it preserves as much of the task-relevant information as possible when propagating through sequential layers [34, 36]. In order to retain this advantage, as shown in Fig. 3, we further extend a single coefficient set to dual coefficient sets: one set $\{\mathbf{\Lambda}_\ell^{\text{MC}}\}$ communicates under the constraint of HMC, while the other $\{\mathbf{\Lambda}_\ell^{\text{LM}}\}$ communicates among different layers.

4.3. Intra-block Enhancement

Since the relation matrix applies a uniform weight distribution across all coefficient elements in the same block, it inherently lacks the ability to apply distinct adjustments for individual elements. To address this limitation, we introduce an intra-block enhancement module that involves a hyper-complex layer (HL) to generate weights $\mathbf{W}$ for a linear projection head parameterized.

Specifically, in HL, the weights are obtained from a suprasphere, which is initialized as orthogonal weights, as shown in Fig. 4. Then, we use the `Hamilton product` $\otimes$ [3] to update the element H of the suprasphere. Define two weights of a element $\widetilde{H_a}$ and $\widetilde{H_b}$ respectively as follows:

$$\widetilde{H_a} = a_0 1 + a_1 j_1 + \cdots + a_{N-1} j_{N-1} \qquad (14)$$

$$\widetilde{H_b} = b_0 1 + b_1 j_1 + \cdots + b_{N-1} j_{N-1}. \qquad (15)$$

Then, we obtain the corresponding updated element W_i via

a hyper-complex layer, which is formulated as:

$$
\begin{aligned}
H =& \widetilde{H_a} \otimes \widetilde{H_b} \\
=& (a_0 b_0 + \cdots + a_0 b_{N-1} j_{N-1})1 + \\
& (a_1 b_0 + \cdots + a_1 b_{N-1} j_{N-1}) j_1 + \\
& \cdots \\
& (a_{N-1} b_0 + \cdots + a_{N-1} b_{N-1} j_{N-1}) j_{N-1}. \quad (16)
\end{aligned}
$$

In the HL, all parameters are hyper-complex numbers, including elements and weights. The `Hamilton product` performs transformations of elements in hyper-complex space, as well as scaling and interpolation between two rotations following a geodesic over a sphere in the $\mathbb{R}^{N-1}$ suprashpere. More details of the forward propagation, the back-propagation, and the parametrization of the hyper-complex layer can be found in supplementary material. Finally, HL uses a real transform $Q : \mathbb{H}^N \to \mathbb{R}^V$ to transform the suprasphere back to the parameter space, and then obtain the corresponding parameter weights $\mathbf{W}$. This is usually achieved by multiple concatenating elements in the suprasphere via a specific rule [11]. By equipping the projection head with HL, the adjustment of individual coefficient elements can be enhanced to achieve a greater impact on the projection direction of the parameter space.

4.4. Overall Architecture

Overall, we develop a SAM-COBOT framework, and for a specific input feature map $\mathbf{M}_\ell$ in the ℓth SAM-COBOT module, the right branch in the SAM-COBOT module produces the adjusted feature map, $\widetilde{\mathbf{M}}_\ell$, formally via:

$$\widetilde{\mathbf{M}}_\ell = \mathcal{F}_\ell(\mathbf{M}_\ell; \mathbf{W}\mathbf{\Lambda}^{\text{MC}}) + \mathcal{F}_\ell(\mathbf{M}_\ell; \mathbf{W}\mathbf{\Lambda}^{\text{LM}}), \qquad (17)$$

where $\mathcal{F}_\ell$ represents ℓ^{th} block of SAM's image encoder.
Fine-tuning. During the fine-tuning phase, SAM-COBOT is fine-tuned in conjunction with the existing PEFT modules. Concurrently, the original components of SAM load their weights from the pre-trained checkpoint, with their parameters remaining frozen.
Loss Function. Following previous works [26, 39], we incorporate a combination of binary cross-entropy loss, denoted as $\mathcal{L}_{\text{ce}}$, and binary dice loss, represented by $\mathcal{L}_{\text{dice}}$, for the fine-tuning of SAM. The overall loss function is derived as:

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \mathcal{L}_{\text{dice}} \qquad (18)$$

5. Experiments

In this section, we evaluate our SAM-COBOT on a diverse range of downstream segmentation tasks. These tasks can be broadly classified into three main categories: (1) Medical image segmentation, (2) Natural image segmentation, and

(3) Remote sensing image segmentation. We begin with a description of the datasets used, followed by the associated evaluation metrics, baseline models, and implementation details. Subsequently, we conduct an ablation study to evaluate the individual contributions of components in our proposed SAM-COBOT. Finally, we compare SAM-COBOT with other prevalent parameter-efficient fine-tuning (PEFT) techniques.

5.1. Experimental Setup

Dataset. We evaluate the performance of our method on 10 datasets. These datasets cover multiple tasks of natural image segmentation (COCO [23], TRCAN [12]), remote sensing image segmentation (NWPU [7–9], SSDD [48], SONAR [37]), and medical image segmentation (ADOME [27], SPLEN [2], MOMO [2], BRAST [1], SEGRAP [35]) .

Evaluation Metrics. In line with previous studies [26, 39], we utilize the Dice Similarity Coefficient (DSC) for evaluating medical image segmentation. For both natural and remote sensing image segmentation, we adopt mean intersection-over-union (mIoU).

Baseline Models. We implement our SAM-COBOT onto two popular PEFT methods for fine-tuning SAM [20], i.e., LoRA [14] and Adaptformer [6].

Implementation Details. In all of our experiments, we employ the ViT-Base version of SAM [20] as our backbone, integrating a box prompt for its prompt encoder input. In line with previous studies [26, 39], we apply a random perturbation to each bounding box, varying between 0 and 50 pixels. Our training employs the Adam optimizer [19]. For medical image segmentation, the initial learning rate is set to 1.25×10^{-6}, and the weight decay is 5×10^{-4} with one image per mini-batch. The number of fine-tuning epochs is set to 25. For natural and remote sensing image segmentation, we follow SonarSAM [39], the initial learning rate is set to 10^{-4}, and the weight decay is 5×10^{-5} with one image per mini-batch. The number of fine-tuning epochs is set to 20. More details are provided in the supplementary material.

5.2. Ablative Studies

Ablation of Main Components. Here, we do an ablation study to show the benefit brought by each component of our proposed SAM-COBOT, i.e., coefficient set (CoS), relation matrix (RM), and hyper-complex layer (HL) on three datasets, including ADOME [27], NWPU [7–9] and TRCAN datasets [12]. We use Adaptformer [6] as the baseline in row 1 of Table 1. Comparing row 2 to row 1, we can see slight performance gains brought by the coefficient set, as it introduces more parameter space to be optimized. Then, solely introducing the hyper-complex layer shows limited improvement as it can only adjust projection

CoS	RM	HL	ADOME	NWPU	TRCAN
✗	✗	✗	90.1	83.0	73.3
✓	✗	✗	90.1 (+0.0)	83.1 (+0.1)	73.4 (+0.1)
✗	✗	✓	90.3 (+0.2)	83.3 (+0.3)	73.5 (+0.2)
✓	✗	✓	90.4 (+0.3)	83.3 (+0.3)	73.5 (+0.2)
✓	✓	✗	90.9 (+0.8)	83.7 (+0.7)	73.9 (+0.6)
✓	✓	✓	91.3 (+1.2)	84.0 (+1.0)	74.1 (+0.8)

Table 1. **Ablation study results (%) on three datasets**: ADOME [27], NWPU [7–9], and TRCAN [12]. The baseline is Adaptformer [6]. Their results are shown in the first row. "CoS": dual coefficient sets. "RM": relation matrix, and "HL": hypercomplex layer.

Strategy of RM	ADOME	NWPU	TRCAN
Fixed	90.2 ± 0.6	83.4 ± 0.6	73.3 ± 0.2
Learnable	91.3 ± 0.5	84.0 ± 0.3	74.1 ± 0.0

Table 2. **Effects of RM on three datasets**: ADOME [27], NWPU [7–9], and TRCAN [12]. The baseline model is Adaptformer [6]. "Fixed": random values. "Learnable": update by back-propagation (i.e., ours).

Linear	HL	ADOME	NWPU	TRCAN
✓	✗	90.9 ± 0.6	83.8 ± 0.2	73.9 ± 0.1
✗	✓	91.3 ± 0.5	84.0 ± 0.3	74.1 ± 0.0

Table 3. **Discussion of hyper-complex layer on three datasets**: ADOME [27], NWPU [7–9], and TRCAN [12]. The baseline model is Adaptformer [6]. "Linear": a linear layer.

directions within each layer. Moreover, the results are further boosted by large margins after introducing the relation matrix, showing its capability to capture interdependencies among different layers. Finally, by integrating the hyper-complex layer, the results reveal clear performance gains, e.g., 1.2% on ADOME dataset.

Effects of Relation Matrix (RM). In Table 2, we demonstrate the efficacy of our proposed learnable relation matrix by comparing it with a fixed matrix initialized with random values. We can observe a significant performance improvement with our method, e.g., 1.1% DSC on ADOME dataset [27].

Linear Layer or Hyper-complex Layer. Table 3 presents a comparison between our proposed hyper-complex layer and a standard linear layer, which is a commonly used module for communicating among channels, i.e., projection directions. The results reveal improvements in performance, e.g., 0.4% DSC ADOME dataset [27]. This suggests that the orthogonality facilitated by the hyper-complex layer plays a beneficial role in enhancing intra-layer communication.

Method	Params(K)	Natural		Remote Sensing			Medical					Avg
		COCO	TRCAN	NWPU	SSDD	SONAR	ADOME	SPLEN	MOMO	BRAST	SEGRAP	
Freeze	0	53.0 ± 0.1	53.9 ± 0.2	59.6 ± 0.9	63.2 ± 0.2	34.5 ± 2.7	23.5 ± 1.2	24.3 ± 11.5	24.3 ± 3.3	60.1 ± 1.4	10.5 ± 0.2	40.7
Lightweight	0	70.4 ± 0.1	70.3 ± 0.2	80.5 ± 0.1	80.2 ± 0.2	79.8 ± 0.1	86.0 ± 0.4	93.4 ± 1.2	86.3 ± 3.0	85.1 ± 0.6	67.8 ± 0.2	80.0
Parameter-efficient Fine-Tuning												
LoRA [14]	147.4	71.8 ± 0.1	72.8 ± 0.1	81.8 ± 0.2	80.7 ± 0.1	82.8 ± 0.1	88.0 ± 0.4	94.4 ± 0.4	86.6 ± 2.2	84.8 ± 0.6	68.7 ± 0.1	81.1
LoRA [14]+Ours	148.3 (+0.9)	**72.1** ± 0.1	**73.1** ± 0.0	**82.5** ± 0.2	**81.2** ± 0.1	**84.6** ± 0.1	**88.7** ± 0.2	**94.9** ± 0.1	**86.7** ± 2.4	**85.3** ± 1.2	**70.1** ± 0.1	**81.8**
Adaptformer [6]	322.7	71.7 ± 0.1	73.3 ± 0.1	83.0 ± 0.1	81.9 ± 0.1	84.1 ± 0.1	90.1 ± 0.2	94.8 ± 0.5	87.6 ± 3.1	85.8 ± 0.2	72.1 ± 0.1	82.4
Adaptformer [6]+Ours	324.0 (+1.3)	**72.2** ± 0.0	**74.1** ± 0.0	**84.0** ± 0.3	**82.4** ± 0.2	**84.9** ± 0.1	**91.3** ± 0.5	**96.4** ± 1.7	**89.2** ± 1.4	**87.3** ± 0.6	**73.1** ± 0.1	**83.6**

Table 4. **Segment anything model (SAM) fine-tuned on a diverse range of downstream segmentation tasks, with the corresponding size of trainable parameters.** All results are based on ViT-Base [10] backbone, and we ignore SAM's lightweight mask decoder when calculating the learnable parameters. We use DSC (%) for medical image segmentation, and mIoU (%) for other tasks as evaluation metrics. "Freeze": without any form of fine-tuning. "Lightweight": freezes all the backbone parameters and only tunes SAM's lightweight mask decoder. "Avg": average.
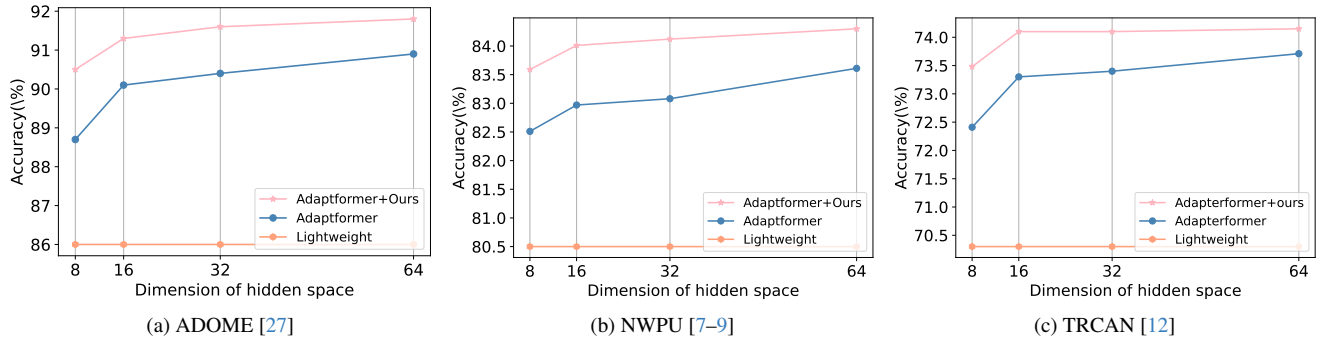

(a) ADOME [27]   (b) NWPU [7–9]   (c) TRCAN [12]

Figure 5. **Results on different dimensions of hidden space r (Best view in color).**

Method	ViT-Base		ViT-Large	
	SSDD	ADMOE	SSDD	ADMOE
LORA [14]	80.7	88.0	81.8	89.1
LORA [14]+Ours	**81.2**	**88.7**	**82.4**	**89.9**
Adaptformer [6]	81.9	90.1	82.1	91.6
Adaptformer [6]+Ours	**82.4**	**91.3**	**82.8**	**93.0**

Table 5. **Results on different backbones.** SSDD [48] and AD-MOE [27] are two datasets we employed.

5.3. Main Results

Comparing to SOTA. We compare our approach against several prevailing PEFT techniques for SAM, including LoRA [14] and Adapterformer [6], on 10 datasets across three domains in the computer vision community. We present their original results and also show our results (by plugging SAM-COBOT in these methods) in Table 4. As shown in Table 4, our SAM-COBOT becomes the new state-of-the-art. Remarkably, our SAM-COBOT, with mini-

mal parameter overhead, significantly enhances both LoRA and Adaptformer. This improvement is particularly noticeable in medical image segmentation. By way of illustration, our SAM-COBOT boosts Adaptformer by 1.2% and 1.0% in terms of mIoU on ADOME [27] and SEGRAP [35] dataset, respectively. Notably, our method achieves 0.5% DSC gains on LoRA on BRAST dataset, despite starting from a lower performance (i.e., 84.8%) than the baseline (i.e., 85.1%). Notably, LoRA achieves a 1.1% improvement through lightweight fine-tuning on average by incorporating 147.4K parameters. In contrast, SAM-COBOT, with a mere addition of 0.9K parameters, achieves an additional 0.7% enhancement in performance. The above results underscore the robustness and generalization capabilities of our SAM-COBOT. More results are provided in the supplementary material.

Qualitative results. Here, we visualize our method's representative example segmentation results against prevailing fine-tuning methods, e.g., LoRA [14] and Adaptformer [6] in five datasets. As shown in Fig. 6, we observe that our approach is able to generalize on diverse scenarios and produce more accurate results.

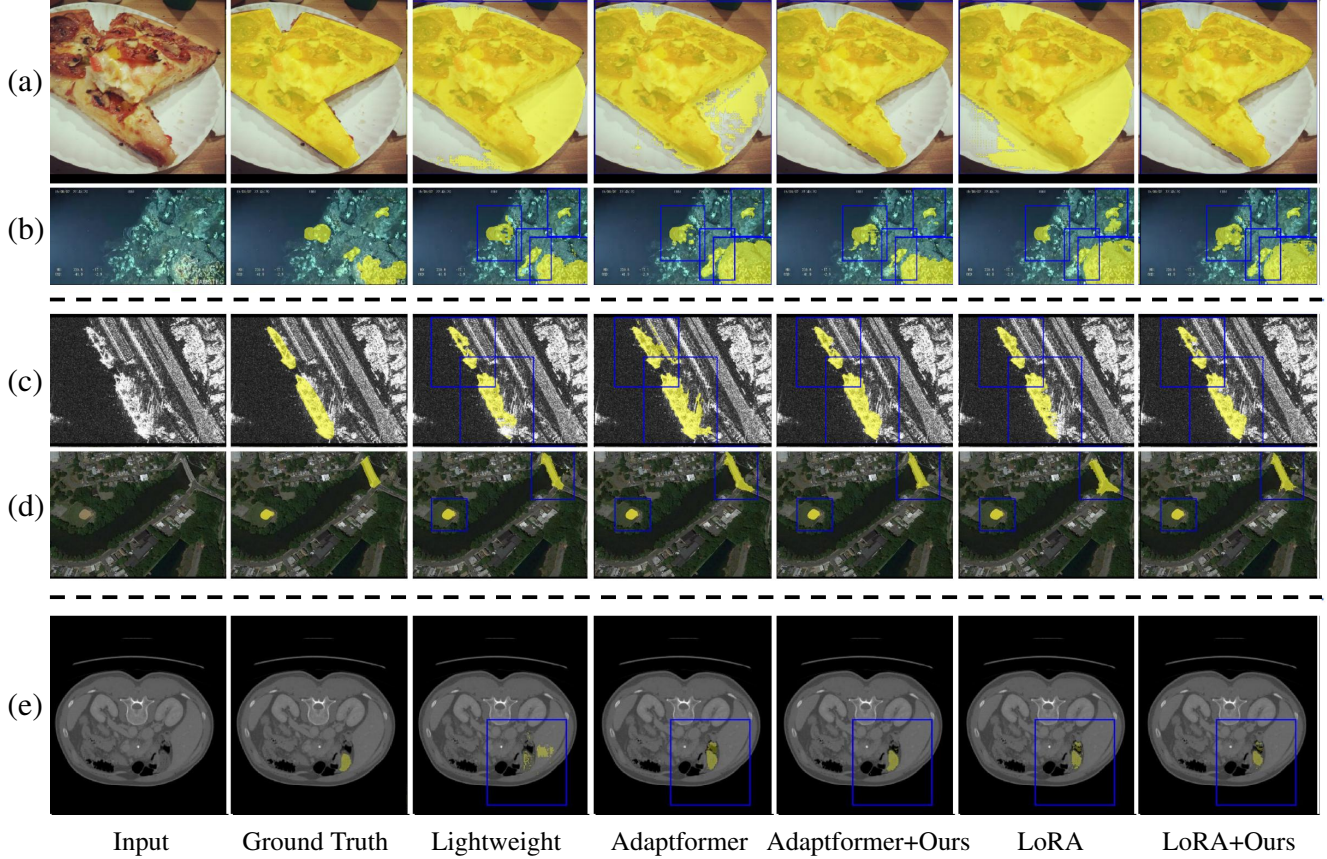**Different Backbones.** We extend our fine-tuning to include

Figure 6. **Qualitative segmentation results on three scenarios**, i.e., (a) natural image segmentation on COCO dataset [23], (b) natural image segmentation on TRCAN [12] dataset, (c) remote sensing image segmentation on SSDD [48] dataset, (d) remote sensing image segmentation on NWPU [7–9] dataset and (e) medical image segmentation on ADOME [27] dataset. "Lightweight": freezes all the backbone parameters and only tunes SAM's lightweight mask decoder.

larger-scale backbones, e.g., ViT-Base and ViT-Large, reinforcing the versatility of our method. This is evaluated on the SSDD and ADOME datasets, where, as Table 5 illustrates, performance improvements are observed consistently. These results demonstrate the generalization of our approach across various transformer architectures.

Different Hidden Dimensions. In Fig. 5, we compare our method with a baseline model, specifically Adaptformer, across various dimensions of the hidden space V. Overall, our method demonstrates distinct advantages in all dimensional settings. It is noteworthy that at lower dimensions, e.g., $V \leq 16$, our method achieves more pronounced performance improvements, exceeding 1.6% on the ADOME dataset [27]. This observation underscores the efficiency of our method in facilitating interaction among bases, particularly when the number of bases, or V, is constrained.

6. Conclusion

In this paper, we equipped PEFT with a cross-block orchestration mechanism to enable the adaptation of the Seg-

ment Anything Model (SAM) to various downstream scenarios, called SAM-COBOT. Specifically, SAM-COBOT introduced a novel inter-block communication module to ensure a comprehensive adjustment of the coefficients for each project direction across the entire parameter space, and an intra-block enhancement module to enhance the coordination of projection directions. By incorporating two modules, SAM-COBOT achieved a proper adjustment of parameter space for new scenarios. Extensive experiments showed that the proposed SAM-COBOT can be easily plugged-and-play and consistently improve two prevalent PEFT paradigms by a large margin across three prevalent scenarios, while only introducing 1K additional parameters.

References

[1] Walid Al-Dhabyani, Mohammed Gomaa, Hussien Khaled, and Aly Fahmy. Dataset of breast ultrasound images. *Data in brief*, page 104863, 2020. 6

[2] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, et al. The medical segmentation decathlon. *Nature communications*, page 4128, 2022. 6

[3] Paolo Arena, Luigi Fortuna, Luigi Occhipinti, and Maria Gabriella Xibilia. Neural networks for quaternion-valued function approximation. In *Proceedings of IEEE international symposium on circuits and systems-ISCAS'94*, pages 307–310. IEEE, 1994. 5

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, pages 1877–1901, 2020. 1

[5] Liyi Chen and Jufeng Yang. Recognizing the style of visual arts via adaptive cross-layer correlation. In *Proceedings of the 27th ACM international conference on multimedia*, pages 2459–2467, 2019. 2

[6] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022. 1, 2, 3, 6, 7

[7] Gong Cheng and Junwei Han. A survey on object detection in optical remote sensing images. *ISPRS journal of photogrammetry and remote sensing*, 117:11–28, 2016. 6, 7, 8

[8] Gong Cheng, Junwei Han, Peicheng Zhou, and Lei Guo. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS Journal of Photogrammetry and Remote Sensing*, 98: 119–132, 2014. 6

[9] Gong Cheng, Peicheng Zhou, and Junwei Han. Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7405–7415, 2016. 6, 7, 8

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 7

[11] Thomas Hawkins. Hypercomplex numbers, lie groups, and the creation of group representation theory. *Archive for History of Exact Sciences*, 8:243–287, 1972. 5

[12] Jungseok Hong, Michael Fulton, and Junaed Sattar. Trashcan: A semantically-segmented dataset towards visual detection of marine debris. *arXiv preprint arXiv:2007.08097*, 2020. 6, 7, 8

[13] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799, 2019. 1

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 1, 2, 3, 6, 7

[15] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727, 2022. 1, 2

[16] Shibo Jie and Zhi-Hong Deng. Fact: Factor-tuning for lightweight adaptation on vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1060–1068, 2023. 1, 2

[17] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, page 2, 2019. 1

[18] Eric Kernfeld, Misha Kilmer, and Shuchin Aeron. Tensor–tensor products with invertible linear transforms. *Linear Algebra and its Applications*, 485:545–570, 2015. 4

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 3, 6

[21] Jungbeom Lee, Jooyoung Choi, Jisoo Mok, and Sungroh Yoon. Reducing information bottleneck for weakly supervised semantic segmentation. *Advances in Neural Information Processing Systems*, 34:27408–27421, 2021. 2

[22] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, pages 109–123, 2022. 2

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6, 8

[24] Gen Luo, Minglang Huang, Yiyi Zhou, Xiaoshuai Sun, Guannan Jiang, Zhiyu Wang, and Rongrong Ji. Towards efficient visual adaption via structural re-parameterization. *arXiv preprint arXiv:2302.08106*, 2023. 2

[25] Wei Luo, Xitong Yang, Xianjie Mo, Yuheng Lu, Larry S Davis, Jun Li, Jian Yang, and Ser-Nam Lim. Cross-x learning for fine-grained visual categorization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8242–8251, 2019. 2

[26] Jun Ma and Bo Wang. Segment anything in medical images. *arXiv preprint arXiv:2304.12306*, 2023. 1, 2, 5, 6

[27] Jun Ma, Yao Zhang, Song Gu, Cheng Zhu, Cheng Ge, Yichi Zhang, Xingle An, Congcong Wang, Qiyuan Wang, Xin Liu,

Shucheng Cao, Qi Zhang, Shangqing Liu, Yunpeng Wang, Yuhui Li, Jian He, and Xiaoping Yang. Abdoment-1k: Is abdominal organ segmentation a solved problem? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6695–6714, 2022. 6, 7, 8

[28] Christian Mattjie, Luis Vinicius de Moura, Rafaela Cappelari Ravazio, Lucas Silveira Kupssinskü, Otávio Parraga, Marcelo Mussi Delucis, and Rodrigo Coelho Barros. Exploring the zero-shot capabilities of the segment anything model (sam) in 2d medical imaging: A comprehensive evaluation and practical guideline. *arXiv preprint arXiv:2305.00109*, 2023. 1

[29] Titouan Parcollet, Mirco Ravanelli, Mohamed Morchid, Georges Linarès, Chiheb Trabelsi, Renato De Mori, and Yoshua Bengio. Quaternion recurrent neural networks. In *International Conference on Learning Representations*, 2019. 2

[30] Zelin Peng, Zhengqin Xu, Zhilin Zeng, Xiaokang Yang, and Wei Shen. Sam-parser: Fine-tuning sam efficiently by parameter space reconstruction. *arXiv preprint arXiv:2308.14604*, 2023. 3

[31] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020. 2

[32] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*, 2020. 2

[33] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30, 2017. 2

[34] Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019. 5

[35] SegRap2023 Challenge. Segmentation of organs-at-risk and gross tumor volume of npc for radiotherapy planning. https://segrap2023.grand-challenge.org/, 2023. 6, 7

[36] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017. 5

[37] Deepak Singh and Matias Valdenegro-Toro. The marine debris dataset for forward-looking sonar semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3741–3749, 2021. 6

[38] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Lst: Ladder side-tuning for parameter and memory efficient transfer learning. *Advances in Neural Information Processing Systems*, 35:12991–13005, 2022. 3

[39] Lin Wang, Xiufen Ye, Liqiang Zhu, Weijie Wu, Jianguo Zhang, Huiming Xing, and Chao Hu. When sam meets sonar images. *arXiv preprint arXiv:2306.14109*, 2023. 2, 5, 6

[40] Yaoming Wang, Bowen Shi, Xiaopeng Zhang, Jin Li, Yuchen Liu, Wenrui Dai, Chenglin Li, Hongkai Xiong, and Qi Tian. Adapting shortcut with normalizing flow: An efficient tuning framework for visual recognition. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15965–15974. IEEE, 2023. 1, 2

[41] Zifeng Wang, Xi Chen, Rui Wen, Shao-Lun Huang, Ercan Kuruoglu, and Yefeng Zheng. Information theoretic counterfactual learning from missing-not-at-random feedback. *Advances in Neural Information Processing Systems*, 33:1854–1864, 2020. 2

[42] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, pages 12077–12090, 2021. 2

[43] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9981–9990, 2021. 2

[44] Dongshuo Yin, Yiran Yang, Zhechao Wang, Hongfeng Yu, Kaiwen Wei, and Xian Sun. 1% vs 100%: Parameter-efficient low rank adapter for dense predictions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20116–20126, 2023. 2

[45] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021. 2

[46] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. Self-regulation for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6953–6963, 2021. 2

[47] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 3

[48] Tianwen Zhang, Xiaoling Zhang, Jianwei Li, Xiaowo Xu, Baoyou Wang, Xu Zhan, Yanqin Xu, Xiao Ke, Tianjiao Zeng, Hao Su, et al. Sar ship detection dataset (ssdd): Official release and comprehensive data analysis. *Remote Sensing*, 13(18):3690, 2021. 6, 7, 8

[49] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 2

[50] Tu Zheng, Yifei Huang, Yang Liu, Wenjian Tang, Zheng Yang, Deng Cai, and Xiaofei He. Clrnet: Cross layer refinement network for lane detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 898–907, 2022. 2